

Jun Rekimoto, Computer Software, Japan, Computer Software Association, November 15, 1994, Vol. 11, No. 6, partial English translation of pp. 65 to 68

“Information Cube: Using Semitransparency in 3D Information
5 Vizualization”

3. Information Cube

“Information Cube” is a three-dimensional (3D) visualization technique employed to represent hierarchical information. It expresses a hierarchical structure by a group of nested cubes. The first level of a hierarchical
10 structure (the root part in a tree structure) corresponds to the outermost cube, and then the next cube placed inside this outermost cube corresponds to the second-level hierarchical information. In this way, a hierarchical structure is converted into and represented by a corresponding nest structure. Each of the cubes is given a label (title
15 character string) attached to its surface to indicate what is contained inside. The final layer of a hierarchical structure is represented by a labeled rectangle instead of a cube.

This cube-type structure is represented as a three-dimensional graphic image by using a visualization system. Users of this system are able to
20 observe the cubic structure by using a typical CRT display or head-mounted display (HMD). The system provides users with an interacting means that can be used for changing visual points, rotating the entire cube by way of DataGlove [19], or closing in to observe detailed information by

selecting a specific part of the cube. This interactive user interface helps the observer's intuitive understanding of three-dimensional images. The surface of each cube is rendered in a translucent color. In accordance with its translucent color, a user can easily identify what is inside the
5 individually colored cubes.

Figure 1 shows an example display when the example given in paper [7] is visualized by using this "Information Cube" representation. Figure 2 shows a visualization example for Unix directories; information contained in 1,500 files (in 50 directories) is visualized in a graphical representation
10 occupying only a single display screen.

The metaphor of the "Information Cube" is quite simply organized so users can easily understand the implication. As cubes (boxes) in the real world are for storing something, this metaphor offers a natural means of presenting information. In addition, a cube can be utilized as an
15 "information container" to store any types of three-dimensional information representation. For example, Figure 3 shows a case where network information is contained in a cube in the form of a 3D graphical presentation. Thus various types of three-dimensional information can be organized into cubic layers.

20 3.1 Translucent Rendering

As shown in Figure 2, the system applies translucent rendering to the surface of each cube. Translucent representation allows users to see the contents of a cube, while giving gradually increasing shields for deeper nesting. If the cubes' surfaces were transparent (i.e., only a wire-frame

presentation is given for cubes), the display contents would be too complicated for users to understand (Figure 4). On the other hand, if surfaces are painted opaque, the second and the following layers are completely hidden from view, so the purpose of hierarchical structure representation cannot be achieved. Translucent rendering can concurrently solve these two conflicting problems. The more information is nested in depth, the more hidden it is from view, but that happens gradually. The complexity of information representation as viewed by users may be controlled within an adequate level.

10 As the concept of “translucency” is common in the real world, users can intuitively understand the meanings of such representation. It may also be possible to associate specific meanings to information by changing the degree of transparency. For example, the cost of accessing information or the importance of information can be associated with the
15 degree of transparency applied. There may be such an application that information with higher access cost may be enclosed in a less transparent cube in order for the users to have an intuitive impression that the associated viewing cost is higher. Translucent rendering may also be used simply for information grouping. For example, Figure 3 shows translucent
20 spheres to represent node groups within a network.

 Translucent rendering itself is not new in the field of computer graphics. Translucent display is also frequently used in volume rendering in the field of visualization. However, this study is considered the first case where a translucent display is used for expressing the information
25 structure itself and also as a means to control the amount of information.

3.2 Object Layout

Some sort of layout (configuration) algorithm is required for converting a hierarchical structure to a nested cubic structure. In general, there are known algorithms for packing as many components as possible
5 into a box, such as for solving the “knapsack problem” or “bottling problem.” Johnson [7] has proposed a two-dimensional algorithm for compact nesting into a box, which is a method of alternately repeating vertically- and horizontally-oriented rectangular layouts.

However, for the reasons given below, the “Information Cube” is not
10 intended for such compact layouts:

- Compact and dense deployment typically requires volume computing; it is not suited for the purpose of instant visualization as sought after in this study.
- Too dense an arrangement of cubes with very small clearance
15 between them would cause difficulty in visually capturing the structure. Rather it is considered necessary to give adequate clearance between cubes in order to give the observer better understanding of its structural image.

Therefore, a two-path algorithm is proposed and implemented for the
20 “Information Cube” concept as shown in Figure 5. In essence, this method calculates the size of each cube bottom-up in the first path. Then, the second path is used for calculating the actual positions of cubes. The distances between cubes can be adjusted by changing some parameters of the algorithm.

On the other hand, if we are not to make the configuration compact, a problem arises such that clearance between cubes can be wasteful, resulting in too big an entire cubic structure. In order to avoid this problem, inner cubes are slightly more scaled down in accordance with nesting
5 levels. Consequently in this method,

- Appearance size is changed by three-dimensional perspective conversion;
- Size is changed by scaling according to nest depths;
- Clear visibility is obtained by translucent rendering.

10 The above three key factors are used for controlling the amount of information displayed on a screen.

```

estimate_size(N: node) {
    if N is the end node or N doesn't have a child, then
        return regular size
15  else
        Sj: Child of node N ( $0 \leq j < n$ )
        for j = 0 .... n - 1 do
            Sj.size = estimate_size(Sj)
            maxsize = Size large enough to contain Sj
20  minsize = 0
        while ( $|max\ size - minsize| > \epsilon$ ) do
            size = (maxsize + minsize) / 2.0
            height = layout_cubes (N, size)
            if (height = < size) then
25  maxsize = size

```

```

        else
            minsize = size
        return maxsize
    }
5
    layout_cubes (N: node, size: float) {
        Stack a child ( $S_i$ ) of node  $N$  on the rectangular area of  $size \times size$ .
        Measure the height
        return height
10 }

    layout ( ) {
        size = estimate_size(topnode)
        layout_cubes (topnode, size)
15 }

```

In the actual algorithm, normal rectangular solids are also allowed as components besides the cubes. Because of this, the algorithm is somewhat more complicated than the one as shown above.

20 Based on some experience, the scaling factor for the inner levels is reduced so that the next nested level is scaled to 90%. That is, when the first level is scaled to 100%, the second level is scaled to 90%, the third level is scaled to 81%, and so on. In addition, when an observer selects a specific cube, his/her viewing point is then moved so that the selected

25 cube will be displayed to occupy the visual area with a viewing angle of

approximately 60 degrees. This feature prevents the inner cubes from being too difficult to observe because of the scaling reduction.

InformationCube: 半透明表示を用いた3次元情報視覚化技法

暦本 純一

3次元情報視覚化においては、ユーザに提示する情報量を適切に制御すること、3次元形状の直観的な把握を助けるための視点移動などの対話技法を充実させることが重要である。本論文では3次元情報視覚化の新しい技法であるInformationCubeを提案する。これは階層情報のための視覚化技術で、ユーザにとって馴染みやすいネストキューブメタファによって情報を表現する。提示情報量を制御するために、半透明レンダリングを情報視覚化に取り入れている。またDataGloveや6自由度入力デバイスを用いた柔軟な対話技法を提供している。本稿ではInformationCubeの視覚化方式、その方式に基づいて構築したシステムの実現手法、および利用経験について述べる。

1 はじめに

巨大なソフトウェアの構造・実行履歴、データベーススキーマ、企業の組織図、巨大プロジェクトのPERT図、交通管制情報など、巨大で複雑な情報を理解し操作する必要性が増してきている。しかし、これらの情報は現状の1次元や2次元の図式では表現しきれないか、現在の技術で得られるスクリーンの解像度では表示できないことがある。

グラフィックスワークステーションの能力が向上し、

一般的なものになりつつある現在、3次元表示を用いて情報を視覚化することは魅力的かつ現実的なアプローチとなりつつある。CTスキャナーなどから採取された数値情報を視覚化する技術はサイエンティフィック・ビジュアライゼーションと呼ばれ、10年以上にわたる研究の蓄積があるが、離散的な情報構造、たとえば階層構造やネットワーク構造、を3次元図形として視覚化する技術はまだ確立しているとは言い難い。現時点でのオープンな問題としては、以下が挙げられるだろう。

- どのような情報構造に、どんな視覚化スキーマが有効であるか。2次元図形の視覚的スキーマには数千年以上の歴史があるが、3次元図形を情報表現の手段として自由に利用できるようになったのはわずか数年前であり、圧倒的に経験の蓄積が不足している。
- どのようにして3次元空間を航行するか。これは単に物理的な視点移動のみならず、情報検索や情報の内容に基づいた移動が含まれる。
- どのようにして3次元情報を操作するか。オブジェクトの選択や移動などの操作をどのようにして行うか。
- どのようにして理解しやすく3次元情報を提示するか。これにはCRTやhead mounted display (HMD)などのデバイスの問題、立体感を増進させるための技法(視差情報、視点移動に追従した画面の更新、オブジェクトの対話的な移動・回転)、提示される情報量の制御の問題などが含まれる。

3次元情報視覚化では、2次元情報にあらたな1次元を追加するなどの方法により、同じ表示面積により多くの情報を提示することが可能である。しかし単純に2次

The Information Cube: Using Semitransparency in 3D Information Visualization

Jun Rekimoto, (株)ソニーコンピュータサイエンス研究所,
Sony Computer Science Laboratory, Inc.

コンピュータソフトウェア, Vol.11, No.6 (1994), pp.63-74.

[論文] 1994年4月21日受付。

元を3次元に置き換えて視覚化を行うと、容易にユーザの理解できる範囲を超えた複雑な視覚情報が生成できてしまい、かえってその有効性が阻害される傾向がある。ユーザの理解できるレベルに提示情報量を抑える技術も併せて研究する必要がある。

また、完全な3次元ディスプレイが存在しない、という前提では、(HMD, CRTを問わず)ある瞬間表示されているのは2次元図形にすぎない。ユーザは視差情報や視点の移動、オブジェクトの回転などの効果によって、提示されている情報を立体として把握している。したがって、従来の2次元図式にも増して、システムが積極的にユーザと視覚情報との対話を支援する必要がある。

しかし、従来の3次元情報視覚化の研究では、これらの課題が明確に意識されてきたとは言えない。本論文では、これらの問題を解決するための方法として、Information Cubeと呼ぶ階層構造向け3次元情報視覚化技術について述べる。これはネストしたキューブによって情報を表現する技法である。提示される情報量を制御する手段として半透明表示の導入を試みている。対話性の問題については、6自由度の入力デバイスとそれを用いた種々の対話技法で対処している。

以下、2節で関連研究について紹介し、3、4節でInformation Cubeとその対話技法について述べる。5節でInformation Cubeの実現技術について述べ、最後に6節で本方式の効果、利用経験について議論する。

2 関連研究

この節では、情報視覚化に関するこれまでの研究事例について簡単に紹介する。

まず当然のことながら、2次元図式を用いた情報視覚化には多くの事例がある(たとえば[8]、[4]など)。しかし、これらの技法の共通の問題点は、複雑な情報を表示するには2次元空間は小さすぎることである。ディスプレイ上にこれらの図式を表示する際には、画面のサイズの制約から、その一部分を切り出して表示するか、全体の縮尺率を上げて無理に画面のサイズに納めることが多い。したがって情報の全体構造と、興味のある部分の詳細情報を同時に得ることは難しい。スクロールやズームなどの対話技術を併用しても、この問題は根源的には解決できない。

Macintoshのファインダ[1]などで用いられているデスクトップメタファは階層情報の視覚化技法と考えることができる。ユーザは、上位レベルのアイコンをクリックしていくことで、下位情報を含むウィンドウを開き、情報にアクセスできる。しかし、多くのウィンドウを開くことで、画面が過度に複雑になりがちである。このため、ウィンドウの位置を整理したりといった無用な操作をユーザに強いることになる。また各階層の関係と開かれたウィンドウとの関係を示す視覚的な対応づけは表示されないの、ユーザが自分で判断しなければならない。アイコンとウィンドウという2種類の表現形式があるが、これにも明確な視覚的な対応づけが見られない、といった問題がある。

3次元図式によって情報を視覚化した先駆的な試みとしてはFairchildらのSemNetがある[3]。これは知識ベースのネットワーク構造を3次元図形として表示したものである。Fairchildらによる経験では、あまり顕著な効果はなかったようだが、これは、システムが提供している対話手段が貧弱で、ユーザが自由に情報にアクセスできなかったこと、また、表示される情報量を制御する手段があまりなく、画面が不必要に複雑になってしまうこと、に起因すると考えられる。

Robertsonらは、階層構造の3次元視覚化技法としてCone Tree[14]を提案している。これは、3次元構造をコーン型の3次元木として表現するもので、各コーンの頂点が階層構造中のノードに、コーンの周辺部がそのノードの子供ノードに対応づけられている。マウスによる対話技法で、興味のあるサブノードを画面の手前に移動することができる。移動の際のスムーズなアニメーションによって、情報構造の理解が促進される。しかし、Robertsonらが自身の論文で指摘しているように、この構造は木構造のうちでも非対称なものに向いており、対称的なものではあまり効果がない。これは、対称的な木構造では、回転しても視覚化表現があまり変化せず、情報構造をユーザが認識する手掛かりに乏しくなるからである。また、図形全体の高さは階層構造の深さに比例しているという点で1次元的であり、子供の数もコーンの円周の長さに1次元的に影響する。したがって、階層構造が深くなった場合、あるいは1段の階層での子供の数が増えた場合、視覚表現が不自然なものになってしまう。

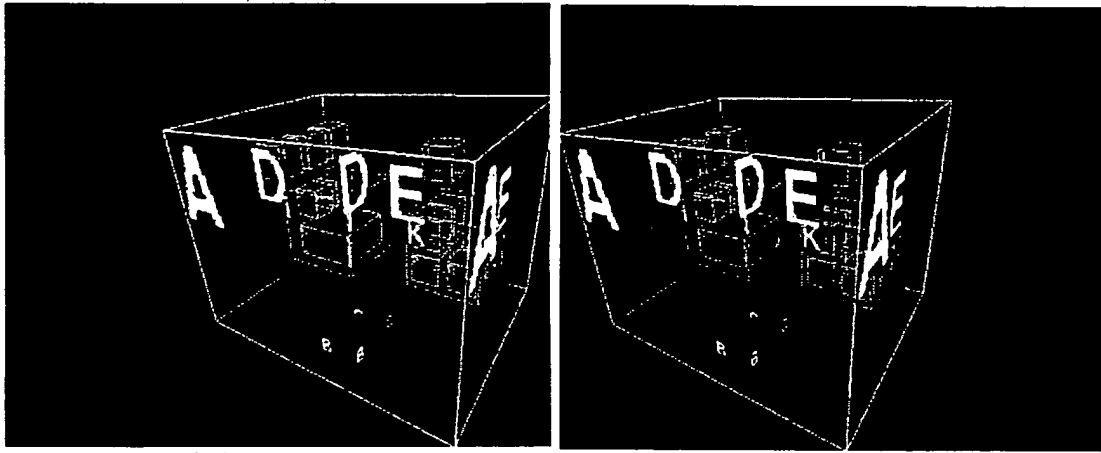


図1 Information Cube (ステレオ画像)

Johnson と Shneiderman の Tree-Map は、階層構造を2次元図形として視覚化する技法である [7]。Tree-Map は稠密に充填された矩形のネストで階層構造を表現している。このスキームによって、階層構造を表現するための無駄な図形 (たとえば木構造におけるノードとノードをつなぐエッジ) なしで、情報を表現できている。しかしながら、Tree-Map の稠密図式は、情報の構造を把握するには不向きである。スクリーンの面積を100%有効に使用しようとするあまり、人間が情報構造を直観的に認識するのに必要な手掛かりが消滅している。

小池はソフトウェアの視覚化において、複数の2次元図式を融合するための手段として3次元視覚化を用いている [21] [9]。たとえばモジュール構成図とシーケンスチャートなど、従来はそれを見る人間が自分の頭の中で概念的に統合していた情報を、3次元表示によって最初から統合された視覚表現としてユーザに提示できる。ただし小池の研究では、情報量が増大した場合の対処法、3次元情報との対話手法については特に言及がない。

3 Information Cube

Information Cube は階層構造情報のための3次元視覚化技法である。ネストしたキューブの集合として階層構造を表現する。階層構造の第1レベル (木構造では「根」となる部分) が最も外側のキューブに相当し、そのキューブの中に置かれたキューブが、第2レベルの情報に相当する、という具合に階層構造をネスト構造に変

換して表現している。各キューブには、その内容を示すためのラベル (タイトル文字列) が表面に貼り付けられている。階層構造の末端に相当する部分は、キューブではなくラベル付きの矩形として表現されている。

このキューブ構造は、視覚化システムによって3次元図形として表示される。ユーザは、普通のCRTディスプレイか、ヘッドマウンテッド・ディスプレイ (HMD) によって、キューブ構造を観察する。システムは視点を変更したり、DataGlove [19]によってキューブ全体を回転させたり、一部分を選択してそこに近づいて詳細情報を表示させたりする対話手段を提供している。この対話性が、3次元図形の直観的な理解を助けている。各キューブの表面は半透明の色でレンダリングされている。したがって、ユーザはキューブの中に何が含まれているかを容易に知ることができる。

図1は文献 [7]で用いられた例を Information Cube として視覚化した場合の画面例である。図2は Unix ディレクトリを視覚化した例で、全ファイル数約1500 (うちディレクトリ数50) の情報が、画面に収まる図式表現として視覚化されている。

Information Cube のメタファは極めて単純で、ユーザはたやすくその意味を理解することができる。キューブ (箱) は、現実世界でも何かを納めるために用いられており、情報表現の手段としても自然なメタファである。また、キューブは任意の3次元情報表現を格納するための「情報コンテナ」として使うことができる。たとえば図



Figure 4

図4 図2と同じ情報を半透明表示を用いずに視覚化した場合
Information Visualization Same as in Fig. 2 but
without Translucent Rendering

3はネットワーク情報を3次元図形として表現しキューブに納めたものである。このように種々の3次元情報をキューブの階層として組織化することができる。

3.1 半透明描画

図2に示されているように、システムは各キューブの表面を半透明でレンダリングしている。半透明表示は、ユーザにキューブの内容を見せると同時に、ネストが深くなるにつれて徐々にその表示を隠している。もしキューブの表面が完全に透明な場合(つまりキューブを単なるワイヤフレームとして表示した場合)、画面表示が複雑になりすぎて、ユーザが理解できなくなってしまう(図4)。一方、表面を不透明な色で表示すると、第2レベル以下が完全に隠されてしまい、階層構造を表現するという目的を達成できない。半透明レンダリングは、この相反する2つの問題を同時に解決している。ネストの深い情報は徐々に隠されているので、ユーザが見る情報表現の複雑さは適度なレベルにコントロールされる。

「半透明」という概念は現実世界に普遍的に存在し、ユーザは直観的にその意味を理解できる。また、透明度を使って情報に意味付けを施すことも可能である。たとえば情報アクセスのコストや重要度などを透明度として表現することができる。アクセスコストの高い情報を透明度の低いキューブで包んでおき、その中を覗くためのコストをユーザに直観的に伝える、などの応用が考えられる。また半透明表示は、単にいくつかの情報をグループ化する目的にも利用できる。たとえば図3は、ネット

ワーク上のノードグループを半透明な球形として表現している。

半透明レンダリング自身はコンピュータグラフィックスの世界では新規なものではないし、視覚化の分野でも、ボリュームレンダリングなどでは半透明表示が多用されている。しかし、情報構造そのものの表現手段として、また情報量を制御する手段として半透明表示を利用したのは本研究が初めてだと考えている。

3.2 オブジェクトの配置

階層構造をキューブのネスト構造に変換するためには、何らかの配置アルゴリズムが必要である。一般的に箱の中に箱をできるだけ稠密に配置するアルゴリズムは「ナップサック問題」あるいは「瓶詰め問題」として知られている。また Johnson は2次元の箱の稠密なネストアルゴリズムとして、各ネストごとに縦配置、横配置を交互に繰り返す方法を提案している [7]。

しかし、Information Cube では、以下の理由により稠密配置は目指さなかった。

- 稠密配置は一般に大量の計算を要し、本研究のように瞬時に視覚化する目的には適していない。
- また、あまりに透き間なくキューブを配置してしまうと、かえって視覚的にその構造を把握することが困難になってしまう。むしろ適度な透き間によって構造を際立たせることが必要であると考えられる。

そこで、Information Cube では図5のような2パスのアルゴリズムを考案し、実装している。この方法の要点は、最初のパスで各キューブのサイズをボトムアップに計算し、次いで第2のパスで実際の位置を計算していくことである。キューブ間の透き間をどの程度空けるか、はアルゴリズムのパラメータを変えることで調整できる。

一方、キューブを稠密に配置しない場合、透き間による無駄のためにキューブの全体が不必要に大きくなってしまいう問題が発生する。これを回避するために、ネストのレベルに応じて内側のキューブの縮尺率を少しずつ上げている。したがって、本技法では

- 3次元の透視変換による見かけ上の大きさの変化
- ネストに応じた縮尺率による大きさの変化
- 半透明表示による表示の鮮明度

の3つの要因によって、画面上の情報量を制御している

```

estimate_size(N: node) {
  if N が末端ノード or N が子供を持たない then
    return 規定サイズ
  else
    Sj: ノード N の子供 (0 ≤ j < n)
    for j = 0...n-1 do
      Sj.size = estimate_size(Sj)
    maxsize = Sj を格納できる充分大きなサイズ
    minsize = 0
    while (| maxsize - minsize | > ε) do
      size = (maxsize + minsize)/2.0
      height = layout_cubes(N, size)
      if (height ≤ size) then
        maxsize = size
      else
        minsize = size
    return maxsize
}

layout_cubes(N: node, size: float) {
  size × size の矩形領域にノード N の子供 (Sj) を積み上げる
  その高さを計測する
  return height
}

layout() {
  size = estimate_size(topnode)
  layout_cubes(topnode, size)
}

```

実際のアルゴリズムでは、構成要素としてキューブだけではなく一般の直方体を許しているの、ここに図示したものよりやや複雑なものとなっている。

図5 キューブ配置

Figure 5 Cube Configuration
ことになる。若干の経験に基づいて、スケーリングは1レベル増すごとに0.9倍になるように設定している。すなわち第1レベルを1とすると、第2レベルは0.9、第3レベルは0.81という具合に縮小されている。また、ユーザが特定のキューブを選択した場合は、そのキューブが視野内に適度な領域(視野角にして約60度)を占めるように視点が移動するので、内側のキューブの縮尺率が高すぎて見えにくくなるような事態は発生しない。

4 3次元対話技法

3次元表示は対話手段によってその価値が大幅に左右される。ある瞬間見えている画像はあくまで3次元図形をある視点から見た2次元図形にすぎず、対話的に視点を変えたり図形を回転させたりする行為によって、3次

元図形の構造をより容易に把握することができるのである。特に、HMDではなく画面を介して情報を観察する場合は、ユーザが対話的にオブジェクトを回転・移動させることでオブジェクトを立体として把握するための視差情報を得ることができる^{†1}。

Information Cube では、対話手段として DataGlove を用いた操作を提供している。DataGlove は、おもにオブジェクトの回転・位置移動、オブジェクトの選択の手段として使われる。

4.1 オブジェクトの回転

オブジェクト全体を、対話的に移動させたり回転させたりする操作は、情報構造を理解する上できわめて重要な役割をはたす。以下にその理由を列挙する。

- 回転や移動による画面の変化(視差)は、両眼視差や視点の変化に伴う動的視差と同様、立体構造を把握する上で重要な情報をユーザに与えている。
- 我々は日常生活で、物品の構造を理解するために、物品を手にとってさまざまな角度からそれを観察するという行為を頻繁に行っている。したがって、同様の操作をコンピュータ内のオブジェクトに対しても行いたいという欲求が強い。
- 回転操作は、情報構造中の興味のある部分を手前に持ってくるため手前にあるオブジェクトによって隠されている部分を見るためにも使える。

Information Cube では、回転の軸は常に選択されているオブジェクトの中心にとられ、DataGlove で「握る」というジェスチャーを与えている間、手をひねるとその角度に対応しただけオブジェクトが回転するようになっている。また、「握る」ジェスチャーを与えている間に手の位置を変化させると、オブジェクト全体の位置もそれに伴って変化する。この2つの操作で、オブジェクトの見たい部分を視野の好きな位置に持ってくることができる(図6)。

回転方向は、垂直の1軸のみに制限している。Information Cube の初期の版では、握った手による自由な回転を許していた。ところが自由度が高すぎたために、ユーザはかえって混乱してしまい、最初のオブジェクト

^{†1} Arthur らの実験 [2]によれば、この情報は静的な両眼視差情報よりも強く立体感に寄与する。

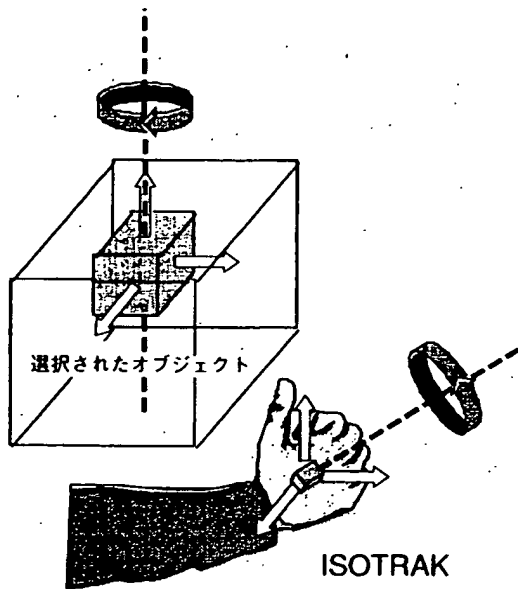
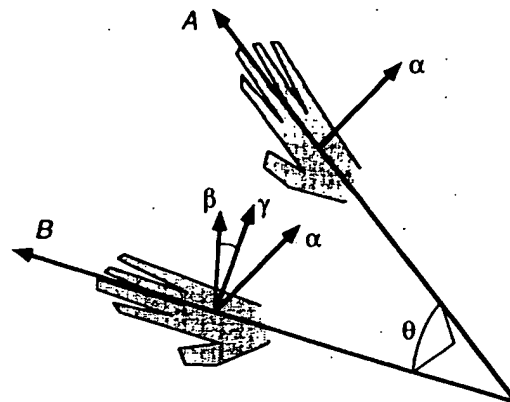


図6 オブジェクトの回転

の方向と、回転後のオブジェクトの方向との対応がつけられなくなることがよく観察された。一方、情報視覚化の目的からすれば完全な自由度の回転は必ずしも必要なく、1軸のみの回転で十分な場合が多い。そこで、DataGloveを装着した手の、上腕方向に関する回転量をスカラー値として取り出し、その値によってオブジェクトを回転させるようにした(図7)。上腕方向の回転量を取り出す理由は以下の通りである。

- 上腕に沿って手を回転させるのは人間の骨格上自然な行為だが、それ以外の軸(たとえば垂直方向)に沿っての回転は不自然な姿勢を強要され、使いやすさが阻害されてしまう。
- また初期の利用経験から、回転の最初と最後で、手の位置が異なる場合が多いことが観察されており、空間に対して固定の軸に関して回転量を取り出すと、手の振れによる変移を回転量として誤認識してしまう、結果的にユーザの意図しない回転が発生してしまう。
- オブジェクトを移動させるための手の位置変化が回転として誤認識されないためには、上腕に関する回転のみを取り出すことが有効である。



α : 回転前の手の甲に垂直なベクトル
 β : 回転後の手の甲に垂直なベクトル
 γ : α の回転成分のみを抽出したベクトル
 A : 回転前の上腕方向
 B : 回転後の上腕方向

$\angle\alpha\beta$ には、上腕方向に関する回転の情報と、上腕の振れに関する動きの情報が含まれている。 α を A と B に垂直な軸によって回転し、補正されたベクトル γ を得る。 $\angle\gamma\beta$ は手の回転の成分のみを持ち、上腕の動きによる誤差は取りのぞかれている。

図7 回転角の補正

この処理によって、オブジェクトを回転させながら位置を変化させるといった複雑な操作がきわめて容易に行えるようになった。すなわち、6自由度の値が得られる位置センサから4自由度のみ(位置に関して3自由度、回転に関して1自由度)しか操作に利用していない。この冗長性がかえって操作性の向上に寄与しているものと考えられる。

4.2 オブジェクトの選択

Information Cubeの階層構造中、特定の場所をさらに詳しく観察するために、システムは「選択」という操作を提供している。これはUnixにおける'cd'(カレントディレクトリ変更)コマンドと本質的に同じで、現在選択されているオブジェクトの子供、兄弟、親をDataGloveで指し示し、「選択」ジェスチャー¹²によって設定する。選択が変更されると、ユーザの視点は選択されたオブジェ

¹² 現在の実現では、「選択」ジェスチャーとして人差し指と中指を同時に伸ばすジェスチャーを用いている。ただし、使用するジェスチャーはユーザが再定義することも可能である。

クトが視野の中に適度な大きさを占めるように自動的に再設定される。

選択による視点の移動は、移動前と移動後の位置関係を見失わないためにスムーズなアニメーションによって表現されている。アニメーション映画などでよく使われている slow-in-slow-out という技法を利用し、視点移動の速度が自然な加速減速によって起きたかのような演出を施している。

選択されたオブジェクトは、他のオブジェクトよりも若干輝度が高めに設定されて視覚的にも強調される。また、前節でも述べたように、回転の軸は選択オブジェクトの中心に設定されているので、回転による移動量からどのオブジェクトが選択されているかが容易に把握できる。Unix の cd と異なり、選択されたオブジェクトの上位レベルのオブジェクト群も視界の一部として表示されており、階層構造全体と、選択されたオブジェクトの関係を把握することが容易である¹³。

選択候補を指し示す手段としては、Liang のスポットライト法 [10] を利用している。これは 3D-CAD システムで部品を選択するために開発された技法で、仮想光線による選択法の拡張版になっている。仮想光線法とは、3次元空間中のオブジェクトを、空間中のある位置から発射する光線と最初に交差したオブジェクトがどれかを調べることで選択する方式である [15] [6]。光線の位置と方向の制御には、ISOTRAK のような 6 自由度を持つポジションセンサを用いる。

仮想光線法は、仮想現実感のデモシステムでよく見受けられる「オブジェクトを直接手でつかむ」方法と比較して、より少ない手の動きで選択できること、手のとどかない場所にあるオブジェクトも選択できること、などの点で優れている。しかし、サイズの小さなオブジェクトや光源から遠距離にあるオブジェクトに光線を当てるには、ポジションセンサの精密な操作が要求され、困難な作業になってしまう。そこで、単純な光線の代わりに、遠くにいくにつれて次第に広がっていく、光源を頂点とした円錐形状のスポットライトを発射し、スポットライ



図8 スポットライトによるオブジェクトの選択

トと交差するオブジェクトを選択するのがスポットライト法である。同時に複数のオブジェクトがスポットライトと交差する場合は、光源からの距離、スポットライトの中心軸からの距離を引数とする優先度関数によって選択すべきオブジェクトを決定する。すなわち、光源により近いもの、スポットライトの軸により近いものが優先度が高くなり、選択対象となる。

Information Cube では、DataGlove の手の甲に位置するポジションセンサからスポットライトを発射させ、その光が最も強く当たっているオブジェクトが選択の候補になる。画面には半透明の光としてスポットライトが表示され、スポットライト光の当たりぐあいによって各オブジェクトの輝度が変わるようになっている (図 8)。Liang のスポットライト法と異なる点は、視点の操作とスポットライトの操作が同時に行えるので、スポットライトの方角を把握しやすい位置に視点を変更できること、選択が容易になるようにオブジェクト全体の方角をあらかじめ変えておけること、である。

5 実装

5.1 システム構成

Information Cube システムは、IRIS 4D VGX ワークステーション上で、MR Toolkit [5] [16] [15] および SGI GL ライブラリ [17] を用いて構築されている。MR Toolkit はアルバート大コンピュータグラフィックス研究所で開発された、仮想現実感を用いたアプリケーションを作成するためのプラットフォームである。各種仮想

¹³ Zuo らは、彼らの階層ネットワークの 2 次元視覚化システムにおいて、部分的なズームによって同様の効果 (全体のコンテキストと着目点の詳細情報の両立) が得られると報告している [20]。

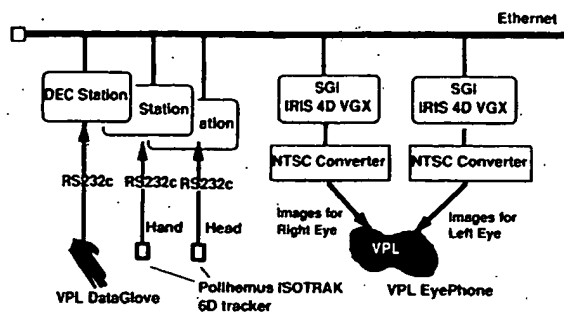


図9 ハードウェア構成

現実感用デバイスの管理、座標系の管理、プロセス間通信の管理などを担当する。実際に画面およびHMDへの描画を担当するのはGLライブラリである。図9に本システムのハードウェア構成を示す。

Information Cubeではテクスチャマッピングや半透明表示のようなレンダリング技法を多用しているが、延べ1500ノード、10レベルの階層構造の視覚表示を毎秒8フレーム程度の頻度で生成可能であった。

リフレッシュレートを維持するために、あるレベル以下の表示は抑制している。たとえば、フォーカスされたノードと直接親子あるいは兄弟関係にないノードに含まれる末端ノードなどは表示が抑制される。しかし、半透明表示による「ぼやけ」の効果が、このような表示の省略を目立たなくしている。

5.2 半透明レンダリング

複数のオブジェクトが半透明物体としてレンダリングされる場合、通常の陰面消去法やZバッファは役に立たない。Zバッファは、ピクセルに奥行き方向の情報(Z値)を付加し、その値を比較することで、手前の物体によって隠された部分の表示を抑制する手法であるが、手前の物体が半透明の場合にも単純に表示を抑制するので、誤った結果が生じてしまう。正しい表示を得るには、Zバッファに加えてさらに別種のバッファを導入するか(たとえば[11])、あるいは各オブジェクトの表示順序をアプリケーションレベルで制御しなければならない。[11]の方法は一般的だがハードウェアでの実現が念頭に置かれており、ソフトウェアのみで実現した場合は効率の低下をきたす。表示順序をアプリケーションで制御する方

法は、通常アプリケーションの複雑化をもたらしてしまう。本システムでは、半透明な物体はすべてネストしているキューブのみ(交差する物体はない)という特性に着目し、アプリケーションによる表示順序制御を行っている。アルゴリズムの詳細は割愛するが、あるキューブを描くには

- キューブの裏側の面
- キューブ内に属するオブジェクトを、視点から遠い順に
- キューブの表側の面

の順にレンダリングすればよく、以下各オブジェクトに対して再帰的に同じ処理を繰り返して表示順序を得る(視点がキューブ内にある場合はこれ程単純にはならないが、同様の方針で制御可能である)。

5.3 文字列表示

キューブ表面に表示される文字列も半透明でレンダリングされている。文字列の大きさ、(3次元空間内での)方向、透明度を任意に設定するために、テクスチャマッピングを利用した文字表示を行っている。

テクスチャマッピングとは、指定したイメージデータ(テクスチャデータ)を、3Dのオブジェクトの表面に貼り付けて表示する技法で、通常は表示されるオブジェクトの迫真性を増すために使われている。ここでは、1枚のテクスチャデータ(現在の実装では256×256テクセル)に表示される文字形状をすべて登録しておき、これをキャラクタジェネレータとして用いている。3次元空間中への文字表示は、このテクスチャデータの一部分を3次元空間中のある領域にマッピングすることで実現できる。マッピングの際の表示属性設定によって、サイズ、方向、透明度などが可変になる。この手法はテクスチャマッピングがハードウェアで実装されているグラフィックス・ワークステーションではきわめて高速に実現でき、情報視覚化のように3次元形状のみならず大量の3次元文字列のレンダリングが必要な分野では有効であろう。

5.4 表示デバイス

Information Cubeは通常のディスプレイとHMDの両方をサポートしている。ユーザは、好みのデバイスを実行時に選択できるようになっている。どちらの場合でも、

入力デバイスは DataGlove を用いている。

HMD を用いた場合は、視点はユーザの頭の動きによって直接コントロールされる。CRT を用いた場合は、ユーザは視点を制御するためのもう 1 つの 6 自由度入力装置 (3SPACE ISOTRAK [12]) を左手に持って操作する。これは Ware の分類 [18] による “eyeball in hand” メタファを使った操作手法である。

6 議論

この節では Information Cube の効果、およびシステム利用の経験により得た知見について議論する。また情報視覚化という観点から、HMD や DataGlove などの機器の実用性について考察する。

6.1 アプリケーション

Information Cube の効果を検証するために、Unix ディレクトリブラウザ、C++ クラスブラウザ、USENET ニュースグループブラウザなどのアプリケーションを構築した。これらのアプリケーションでは、視覚化部は共通のプログラムで、実データを階層構造に変換する部分のみが異なる。この他に考えられるアプリケーションとしては、Gopher など、階層型に情報が格納されたオンライン情報サービスへのユーザインタフェース、企業内の情報システム、電子図書館などが考えられる。

また、複数の階層情報から成る世界を渡り歩くために、都市のメタファを利用した世界を構築した。この世界 (情報都市) では、各情報を具現化したキューブがあたかも都市におけるビルのように林立し、ユーザはその間を自由に移動することができる。半透明の情報から、ユーザはキューブ内に格納されている情報を外部から推測することができる。目的のキューブに容易に接近するために、NaviGrid という技法を用いている [13]。DataGlove から発射される仮想光線を、都市の地面に相当する部分に発射して目的地を設定する方式で、近いところでの精密な位置設定と、遠いところへの高速な移動を両立させることができる。

6.2 Information Cube の特質

実際にプロトタイプシステムを利用して、特に全体構造の表示と細部表示とが連続的に結合されている点に本

方式の優位性を感じた。従来の 2 次元による情報表示では、レベルの異なる情報を連続的に表示することは困難であるが、本方式では、視点の 3 次元空間中の移動によって全体と部分が自然に結合されている。一般に 3 次元空間中の移動操作は複雑になりがちであるが、本方式では、オブジェクトの選択に付随して視点の位置を自動的に変化させることでこの問題に対処している。また、半透明表示により、全体を見渡したときの眺めも過度に複雑にはならず、情報構造を大局的にとらえるために有効であった。したがって、本方式は (視点移動などの対話を通して) 情報の全体構造を直観的に理解するときに、特に有効であると考えている。

また、ユーザが、現在着目している部分が全体の中でどこに位置付けられているのかを知るために、「一歩さがって全体を見渡す」という視点移動操作を頻繁に行うことが観察された。現実世界においても、対象との位置をいろいろ変化させて理解を深めることは自然に行われており、本システムのユーザが現実世界と近い方法で情報構造を把握しようとしているものと考えられる。

Cone Tree と比較した場合、Information Cube では対称的な木構造からも識別しやすい 3 次元形状を生成することができる。これは箱を 3 次元空間に組み上げていくこと、箱の内容に応じて、各箱の縦横比が異なるので、外部から見たときに各々の箱を識別しやすくなること、などの要因によるものである。

また、Information Cube は階層の深さや子供の数に対してより柔軟に対応できる。階層が深くなった場合の問題は、内部のキューブほど縮尺率が高くなっていることで緩和されている。Cone Tree と異なり、子供はキューブ内部に 3 次元的に配置しているので、見掛け上の大きさよりも格納数が多い。たとえばコーンの円周に (1 次元的に) 1000 個のオブジェクトを配置するよりも、キューブ内部に $10 \times 10 \times 10$ 個のサブキューブとして配置するほうが自然な結果を得ることができる。

6.3 表示技法

半透明表示によって内部を段階的に隠蔽したことの代償として、内部情報を詳しく見るには対話的にキューブの中を航行していかなければならない。これはユーザに過度な対話性を要求し問題であるように見える。しかし

この問題は、3次元図形の理解は対話的な操作により促進されること、半透明表示の有無に係わらず、遠方にある情報にアクセスするためには航行機能は不可欠であること、などの理由を考えると許容できるものと考えられる。

画面に提示される情報量を制御する表示技法としては、半透明の他にフォグ(遠くの物体を霞ませて表示する技法)、デプスキュー(距離に応じて物体の色を変化させる技法)などがある。すでに図3によるネットワーク図示などで、デプスキューにより立体構造の把握が容易になることが観察されている。これらの技術の本格的な評価は今後の課題である。

このような、いわば情報表示のリアリティを増すための技法と共に、情報を認識しやすくするためにあえて現実に反した表示を行うことも重要であるとの知見を得た。たとえば Information Cube では各キューブの輪郭は明度を上げて強調されており、識別性を向上させている。またキューブ手前と裏側では透明度が異なり、キューブの内側と裏側の区別が容易にできるようになっている。またキューブの内部に視点が移動した場合は、着目しているキューブを別のキューブが遮らないように、視線と交差するキューブは一部表示を抑制している。このような、いわば細かな magic を積み重ねていくことも情報の理解度を向上させるためには無視できない要因であると考えている。

6.4 HMD と CRT の比較

Information Cube は、全く同一の情報に対して CRT と HMD によるアクセスを許しているので、両者の優劣に関しての知見を得ることができた。

本システムで利用している HMD (VPL EyePhone) は民生用の液晶テレビを用いており、解像度の点で情報視覚化には適しているとは言いがたい(単純計算で、CRT の約 1/20 のピクセル数しかない)。また、色の解像度が劣る点も問題(HMD 内のレンズによって画像が拡大されており、3原色が個別のピクセルとして見えてしまう)で、半透明物体中の半透明物体など、微妙な色でレンダリングされたオブジェクトの識別が困難であった。

一方、対話性の点では HMD は CRT に優る点があった。特に、CRT を介した操作では、現在の視点の位置

や視線の方向を喪失しやすいこと、スポットライトによる選択において、光線の方向を制御しにくいこと、が頻繁に観察された。このとき、HMD による両眼視差の効果よりは、視点の移動に追従した画像更新による効果のほうが顕著に操作性に寄与することが主観的に感じられた¹⁴。

以上のような観察から、3次元情報視覚化に関しては、(比較的大きな)スクリーンで、頭の移動に対応して画像の視点を変化させるような、両者を融合させたような表示方式が最も望ましいと考えている。

7 おわりに

この論文では、3次元情報視覚化に関する新しい技法である Information Cube について述べた。その内容は、おもに以下の3点である。

- ネストされたキューブを用いた自然で親しみやすい視覚化スキーマ
- 視覚提示の情報量を程よく制御するための半透明レンダリング
- 情報の理解を促進するための3次元対話技術

この方式に基づいて、実際に3次元視覚化システムを構築し、いくつかのアプリケーションに適用した。現時点では、本方式の優位性を示す定量的データは得られておらず、今後の課題であると考えている。非公式な観察を通じて、本方式の実用性が確認でき、特に提示情報量の制御と対話技術の充実が情報視覚化の理解を促進することが、新しい発見であったと考えている。

謝辞

本論文は、私がアルバート大コンピュータグラフィックス研究所に客員研究員として滞在している間に行った研究に基づいている。本研究を進めるにあたり、指導して頂いた Mark Green 教授に感謝する。Chris Shaw, Jiandong Liang との議論、彼らからの助言は大変有益であった。MR Toolkit を習得する上で、Dani Beaubien を始めとするアルバート大コンピュータグラフィックス研究所の所員からさまざまなアドバイスや支援を受けた。最後に、本研究の機会を与えて頂いたアルバート大と日本電

¹⁴ HMD 使用時に、意図的に左右の目に(視差なしの)同じ画像を表示しても、特に大きな違和感は感じられなかった。

気株式会社に感謝する。

参考文献

- [1] Apple Computer, Inc.: *Macintosh User Interface Guidelines*, Addison Wesley, 1992.
- [2] Arthur, K. W., Booth, K. S., and Ware, C.: Evaluating 3D Task Performance for Fish Tank Virtual Worlds, *ACM Transactions on Information Systems*, Vol. 11, No. 3 (1993).
- [3] Fairchild, K. M., Poltrock, S. E., and Furnas, G. W.: *SemNet: Three-Dimensional Graphic Representations of Large Knowledge Bases*, Lawrence Erlbaum Associates, 1988, pp. 201-233.
- [4] Gansner, E. R., North, S. C., and Vo, K. P.: DAG - A Program that Draws Directed Graphs, *Software - Practice and Experience*, Vol. 18, No. 11 (1988).
- [5] Green, M. and Beaubien, D.: *Minimal Reality Toolkit Version 1.2 Programmer's Manual*, Department of Computing Science, University of Alberta, Edmonton, Alberta, 1992.
- [6] Jacoby, R. H. and Ellis, S. R.: Using virtual menus in a virtual environment, *Proceedings of the Symposium on Electronic Imaging, Science and Technology*, 1992.
- [7] Johnson, B. and Shneiderman, B.: Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures, *IEEE Visualization '91*, 1991, pp. 284-291.
- [8] Kamada, T.: *Visualizing Abstract Objects and Relations*, World Scientific, 1989.
- [9] Koike, H.: The Role of Another Spatial Dimension in Software Visualization, *ACM Transactions on Information Systems*, Vol. 11, No. 3 (1993).
- [10] Liang, J. and Green, M.: Geometric Modeling Using Six Degrees of Freedom Input Devices, *Proc. of the 3rd International Conference on CAD & Computer Graphics (CAD/Graphics '93)*, Beijing, China, 1993.
- [11] Mammen, A.: Transparency and Antialiasing Algorithms Implemented with the Virtual Pixel Map Technique, *CG&A*, Vol. 9, No. 4 (1989), pp. 43-55.
- [12] Polhemus, Inc.: *3SPACE ISOTRAK User's Manual*, Colchester, Vermont, 1987.
- [13] Rekimoto, J. and Green, M.: The Navigation Grid: An Effective Technique for Virtual Space Navigation, *Proceedings of the Fifth Annual Western Computer Graphics Symposium*, Vernon, British Columbia, Canada, Mar. 1993.
- [14] Robertson, G. G., Mackinlay, J. D., and Card, S. K.: Cone Trees: Animated 3D Visualizations of Hierarchical Information, *Proceedings of ACM CHI '91*, 1991, pp. 189-194.
- [15] Shaw, C., Green, M., Liang, J., and Sun, Y.: Decoupled Simulation in Virtual Reality with the MR Toolkit, *ACM Transactions on Information Systems*, Vol. 11, No. 3 (1993).
- [16] Shaw, C., Liang, J., Green, M., and Sun, Y.: The Decoupled Simulation Model for Virtual Reality Systems, *CHI '92 Conference Proceedings*, 1992.
- [17] Silicon Graphics, Inc.: *Graphic Library Programming Guide*, Mountain View, California, 1991.
- [18] Ware, C. and Osborne, S.: Exploration and Virtual Camera Control in Virtual Three Dimensional Environments, *Proceedings of the 1990 Symposium on Interactive 3D Graphics*, 1990, pp. 175-183.
- [19] Zimmerman, T. and Lanier, J.: A Hand Gesture Interface Device, *Proceedings of CHI and Graphics Interface 1987*, 1987, pp. 189-192.
- [20] Zuo, Z., Dill, J., and Bartram, L.: Variable Zoom: A Fisheye View of Hierarchical Networks, *Proceedings of the Fifth Annual Western Computer Graphics Symposium*, 1993.
- [21] 小池英樹, 石井威望: 3次元ソフトウェア視覚化の枠組と実例による有効性の評価, 情報処理学会論文誌, Vol. 33, No. 6 (1992).